

HYPERNEURONS. SPECTRAL SYNTHESIS OF NEUROSTRUCTURE MODULES

Alexander N. Sychev

Riga, Latvia

Abstract

In the synthesis of the hyperneuron module, lies the mathematical idea of a tangent bundle in the space of a logical function definition. The base of the tangent bundle and the input affine transformations of the neuron are determined on the basis of Walsh spectral representations. The example of synthesis for a randomly given logical function of 6 variables is presented.

Key words: *tangent bundle, neuron, Walsh spectral representations*

1. INTRODUCTION

It is obvious that an abundant set of neurons of the same type though endowed with some orderliness of connections, represents itself as a device with little defined functionality. So, automatically, one may come across the existence of operational errors, and introduce into consideration a hypothetical surface on which to search the global minimum of failure using the roving method. But if to allow, as in many other devices for information transformation, the use of preliminary filtering of the incoming signals, then the tasks solved by the neural network are simplified greatly. As a result, the quality of the solution increases and that manifests itself both in terms of reducing errors and in the appearance of a neural structure that will be convenient both for synthesis and analysis.

For this purpose, this paper considers the application of linear filters at the inputs of logical devices (LU). In order to carry out the task the apparatus of Walsh spectral representations is used. In addition to the input binary variables x_i , $i = 1 \div n$, from the multitude $\{x_1, x_2, \dots, x_n\}$ any of the 2^n linear combinations can be used. Mathematically, this is described by the expression $x' = \sigma \otimes x$, where σ - binary $(m \times n)$ -matrix, representing the transformation of the input argument x , $\otimes$ - the operation of binary multiplication in the field of deductions $GL_n(2)$, $x' = (x'_1, x'_2, \dots, x'_m)$ is the result of transformation: $x'_i = \sigma_{i1}x_1 \oplus \sigma_{i2}x_2 \oplus \dots \oplus \sigma_{in}x_n$, $i = 1 \div m$, $\oplus$ - the operation of addition in the field $GL_n(2)$. In such a way, the transformed variable x'_i represents a linear filter at the i -input of LU.

The application of such filters leads to the expansion of LU functionality. For instance, if some neuron implements a logical function $g(x)$, then the transformation σ of signal x at the input, insures the implementation of the function $g^\sigma(x) = g(\sigma \otimes x)$ by the same neuron. The power of the multitude of various transformations is quite great. So, for non-degenerate σ dimensions $n \times n$ it amounts to

$Q = \prod_{i=0}^{n-1} (2^n - 2^i)$. In this respect there appears an interest in a variant of a simple implementation of a complex function $f(x)$, for example, as a sum of neural functions $g_i(x)$ of a series of k neurons with the input transformation σ_i , $i = 1 \div k$:

$$f(x) = g_1^{\sigma_1}(x) \oplus g_2^{\sigma_2}(x) \oplus \dots \oplus g_k^{\sigma_k}(x).$$

However, with the increase in the dimension n the value Q lags behind the number F in different logical functions significantly: $F = 2^N$, $N = 2^n \xrightarrow{n \uparrow} Q \ll F$. As a result, the length k of the neural row increases, and in this way, naturally, both technical and virtual (programming) expenses on the realization also increase.

The extension of the transformation abilities at the input can be achieved by means of affine transformations $\alpha: x^\alpha = \sigma \otimes x \oplus \eta = x' \oplus \eta$, where $\eta = (\eta_1, \eta_2, \dots, \eta_n)$ - is the binary vector of the argument $x' = \sigma \otimes x$ inversion. The number of different inversions is defined by value 2^n . As a result, the power of the set of input transformations increases to the value $2^n \times Q$. But, as it has been pointed out in [1], there won't be qualitative increase in the efficiency of such an implementation. There also won't be an expected improvement in case of using transformations σ in dimension $m \times n$, where $m > n$.

But it should be noted that the use of affine transformations gives rise to the idea of splitting the domain (space) of the definition X logical function (LF) $f(x), x \in X$, into non-intersecting fragments (spaces): $X = X_1 \cup X_2 \cup \dots \cup X_p$, $X_i \cap X_j = \emptyset$, $i, j = 1 \div p$. In this case, there appears a possibility of the economical realization $f(x)$ by means of joint realization of the family $\{f^{(i)}(x)\}$ partial LF $f^{(i)}(x) = f(x) | x \in X_i, i = 1 \div p$, induced by the splitting of X . An analogue of this method could be found in the work [2]. The efficiency of this idea is obvious, if we put forward the idea of the implementation of the whole family $\{f^{(i)}(x)\}$ by means of one neuron with one $m \times n$ -transformation σ and the family of the corresponding inversions $\{\eta^{(i)}\}$, $\eta^{(i)} = (\eta_1^{(i)}, \eta_2^{(i)}, \dots, \eta_m^{(i)})$, $i = 1 \div p$. In this case, the multitude of variants of input linear transformations becomes no less than $2^{p \cdot m} \times Q$.

The purpose of this article is a general description of the spectral method for the synthesis of the custom neural module, the structure of which contains a neuron and a device for implementation of the affine transformations in the family $\{\alpha_1, \alpha_2, \dots, \alpha_p\}$, $\alpha_i = \sigma \otimes x \oplus \eta^{(i)}$, $i = 1 \div p$, of the input signal x .

2. SOME NOTES FOR UNDERSTANDING SYNTHESIS OF A HYPERNEURON

It is obvious that an arbitrary partition of function $f(x)$ space may be technically difficult for its realization and won't bring about the expected effect. Therefore, it is reasonable to apply some ordering rules.

Firstly, the function definition space should be subdivided into subspaces equal in size in order to simplify the technical (including software) implementation of the module. For example, when dividing X into two subspaces, one might use one variable, some linearized $x'_i = \sigma_{i1}x_1 \oplus \sigma_{i2}x_2 \oplus \dots \oplus \sigma_{in}x_n$. So, in case x'_i we shall get $X = X_0 \cup X_1$, where $X_0 = \{x : x'_i = 0\}$, $X_1 = \{x : x'_i = 1\}$, wherein the powers of subspaces are equal to $|X_0| = |X_1| = 2^{n-1}$.

Secondly, the function $g(x)$ of the only module's neuron should be efficiently adapted by the affine transformations α_i to the corresponding functions $f^{(i)}(x)$ of the family $\{f^{(i)}(x)\}$. Consequently, it is desirable to ensure, if not the equality of the ranks $r_i = |f^{(i)}(x)|$ of the family functions, then, at least, their closest proximity, $r_i \cong r_j$, $i, j = 1 \div p$.

The implementation of these terms is feasible by analyzing the spectrum s of the original function $f(x)$ in the representations of the basis of Walsh functions [3]. It is necessary to note that argument ω of the coefficient $s(\omega)$ of the spectrum s triggers linearized variable $x'_\omega = \omega_1x_1 \oplus \omega_2x_2 \oplus \dots \oplus \omega_nx_n$. Therefore, the value of the coefficient $s(\omega)$ expresses the

difference of single values $f(x)$, placed in the corresponding subspaces X_0 и X_1 :
 $s(\omega) = k_0 - k_1$, $k_0 = |f^{(0)}(x)| = |f(x)| : x \in X_0$, $k_1 = |f^{(1)}(x)| = |f(x)| : x \in X_1$,

as it is predetermined by the Walsh function definition. That is why, in order to ensure the closeness of ranks k_0 and k_1 of the functions $f^{(0)}$ and $f^{(1)}$, generated in the bundle of space $X = X_0 \cup X_1$, it is necessary to highlight coefficients $s(\omega)$ in the spectrum s with the minimal absolute value $|s(\omega)| = \min\{|s(\omega)|\}$.

Similarly, it is possible to make a division of X into 2^t in subspaces, using t of linearized variables $x'_{\omega 1}, x'_{\omega 2}, \dots, x'_{\omega t}$. Having this aim in view, for example, at the stage where $(k+1)$, where $k < t$, where multitude of subspaces $\{X_i\}, i = 0 \div (2^k - 1)$ have been defined, and also a set of corresponding partial functions $\{f^{(i)}(x)\}$, it is necessary to calculate spectra $s^{(i)}(\omega)$ of these functions and their total density $D = \sum_i |s^{(i)}|$, or $D = \{d(\omega)\}$, where $d(\omega) = \sum_i |s^{(i)}(\omega)|$. It is also necessary to define a linear space M_k , generated by arguments $\omega^1, \omega^2, \dots, \omega^k$, and set in D the values $d(\omega) = 0$, if $\omega \in M_k$. Then, in order to select a variable $x'_{\omega(k+1)}$, it is needed to highlight in D a minor set of coefficients where $D^m = \{d^m(\omega)\}$, $d^m(\omega) = \min D$. After that, one of the coefficients $d^m(\omega^b)$ is allocated in D . According to the argument ω^b of this coefficient $d^m(\omega^b)$, $\omega^{k+1} = \omega^b$ is allocated and a corresponding linearized variable $x'_{\omega(k+1)} = \omega_1^{k+1} x_1 \oplus \omega_2^{k+1} x_2 \oplus \dots \oplus \omega_n^{k+1} x_n$. Continuing further on, variables $x'_{\omega 1}, x'_{\omega 2}, \dots, x'_{\omega(k+1)}$ define subspaces $X_i, i = 0 \div (2^{k+1} - 1)$, wherein, $X_i = \{x : (x'_{\omega 1}, x'_{\omega 2}, \dots, x'_{\omega(k+1)}) = i\}$.

Thus, mathematically speaking, a tangent bundle is formed along the base M in space X ; this base represents a linear space generated by vectors $\omega^1, \omega^2, \dots, \omega^{k+1}, \dots, \omega^t$. Moreover, the subspace X_i is the tangent space to M at the point where $i = (x'_{\omega 1}, x'_{\omega 2}, \dots, x'_{\omega t})$, and represents itself as i -th layer in the tangent bundle.

For partially defined LF $f(x)$ the principle of stratification X is similar. The difference lies in the necessity of equal power stratification both in the region $X_1^* = \{x : f(x) = 1\}$, and in the region $X_0^* = \{x : f(x) = 0\}$. For this purpose, fully defined LF f_1^*, f_0^* should be set accordingly:

$$f_1^*(x) = \begin{cases} 1, & \text{if } x \in X_1^*, \\ 0, & \text{if } x \notin X_1^* \end{cases}, \quad f_0^*(x) = \begin{cases} 1, & \text{if } x \in X_0^*, \\ 0, & \text{if } x \notin X_0^* \end{cases} \quad \text{and consider the corresponding}$$

spectra s_1^*, s_0^* .

3. STRUCTURE OF THE HYPERNEURON

Stratification of the space X into a set 2^t of subspaces $X_i, i = 0 \div (2^t - 1)$ is easy to carry out technically on the basis of applying a linearized decoder (Fig.1). This is a **DC** decoder, the inputs of which are connected to the outputs of the linear conversion unit σ_{dc} . This unit also performs multiplication $\sigma_{dc} \otimes x$ where the binary matrix of the dimension $t \times n$ has the form

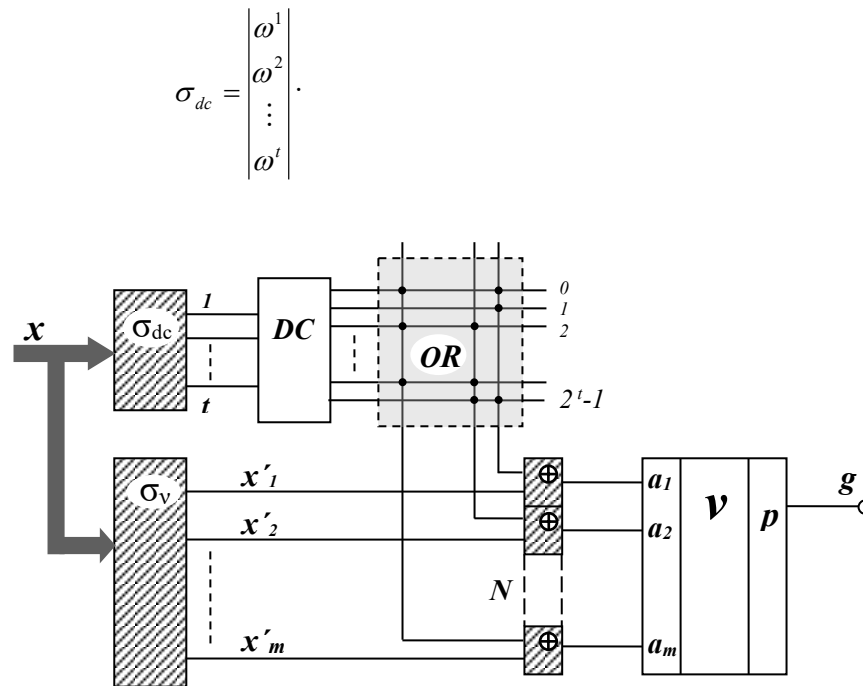


Fig. 1. General structural diagram of a hyperneuron:
v – a neuron, **DC** – a decoder, **N** – a line of adders according to *mod2*,
 σ_v and **σ_{dc}** – blocks of linear decoders of a neuron and a decoder,
OR - matrix of connections “OR”.

The outputs of the decoder represent i tangent points on the base M , and the numbers i layers of the tangent bundle. In order to realize fragment $f^{(i)}(x)$, associated with layer X_i , it is necessary to make an affine transformation $\alpha_i = (\sigma_v, \eta^{(i)})$ of the type $x^{\alpha i} = \sigma_v \otimes x \oplus \eta^{(i)} = x' \oplus \eta^{(i)}$. Having this purpose in view, the **DC** outputs are connected via **OR** matrix with the first inputs of *mod2* adders in the **N** line according to the following rule. The output of the decoder with the number i is connected to the first input of the adder, which is placed on the j -th input of the neuron, if in the vector of inversion $\eta^{(i)} = (\eta_1^{(i)}, \eta_2^{(i)}, \dots, \eta_m^{(i)})$ the component $\eta_j^{(i)} = 1$, and if not connected otherwise. The second inputs of the adders in **N** are connected with the outputs of the linear transformation block σ according to the correspondence rule: – j -th output of the block σ , that generates signal x'_j , is switched up to the adder at j -th neuron's input.

4. EXAMPLE OF HYPERNEURON SYNTHESIS

Let in n -dimensional binary space $L_n(2)$, LF $f(x)$ with a uniform probability distribution density be given randomly. For the clarity of perception let $n = 6$, $k = |f| = 31$. The function f and its spectrum s are represented by Karnaugh tables in Fig.2.

In spectrum s , in accordance with the explanations, among coefficients $c |s(\omega)| = 1$ two coefficients $s(\omega^1)$ and $s(\omega^2)$ are chosen, arguments ω^1 and ω^2 of which are obviously independent linearly. The selected coefficients are circled in the table s . Coefficient arguments possess vector values

$\omega^1 = 101000, \omega^2 = 111010$. This implies that $\sigma_{dc} = \begin{vmatrix} \omega^1 \\ \omega^2 \end{vmatrix} = \begin{vmatrix} 101000 \\ 111010 \end{vmatrix}$. The corresponding linearized variables $x'_{\omega 1} = x_1 \oplus x_3$ and $x'_{\omega 2} = x_1 \oplus x_2 \oplus x_3 \oplus x_5$ represent a vector basis generating a simple base M , containing 4 points: $\{(x'_{\omega 1}, x'_{\omega 2})\} = \{(00), (01), (10), (11)\}$, or $\{(x'_{\omega 1}, x'_{\omega 2})\} = \{0, 1, 2, 3\}$, which is the same. The stratification of space $L_6(2)$ by these points is shown in Karnaugh table **DC**.

In mathematics, the tangent layer to point x on the base M is usually denoted by the symbol $T_x M$. The following four Karnaugh tables in Fig.2 represent fragments, $f^{00}, f^{01}, f^{10}, f^{11}$, - values of the original function f on the relevant layers: $T_{00}M, T_{01}M, T_{10}M, T_{11}M$.

The method of spectral synthesis of a single neuron ν with the corresponding affine transformations at the input $(\sigma_\nu, \eta^{(00)}), (\sigma_\nu, \eta^{(01)}), (\sigma_\nu, \eta^{(10)}), (\sigma_\nu, \eta^{(11)})$ for the functions $f^{00}, f^{01}, f^{10}, f^{11}$ on the fragments formed by such a bundle is described in the works [2,4].

--

Fig. 2. Karnaugh tables, representing:
f – function $f(x)$; **s** – spectrum of the function;
DC – value of vector (x'_1, x'_2) , generating bundle on the base M ;
T₀₀M, T₀₁M, T₁₀M, T₁₁M – tangent spaces with value **f**.

It should be noted that the exact implementation of the given function $f(x)$ could be achieved but it may lead to the increase in the dimension of matrix σ_ν . Therefore, the aim of synthesis here is the formation of a hyperneuron with the function $g(x)$ that approximates $f(x)$ quite accurately, taking into account acceptable costs on linear transformations of σ_ν . Matrix σ_ν , determined as a result of the synthesis, has the dimension 9×6 . In accordance with the chosen spectral arguments $\omega^1, \omega^2, \dots, \omega^9$ the matrix looks like this:

$$\sigma_v = \begin{vmatrix} \omega^1 \\ \omega^2 \\ \omega^3 \\ \omega^4 \\ \omega^5 \\ \omega^6 \\ \omega^7 \\ \omega^8 \\ \omega^9 \end{vmatrix} = \begin{vmatrix} 001110 \\ 001111 \\ 001000 \\ 011010 \\ 000100 \\ 110000 \\ 000101 \\ 011101 \\ 001011 \end{vmatrix}.$$

Consequently, the second outputs of the adders $m2$ in the line N receive the converted signals:

$$x'_{\omega 1} = x_3 \oplus x_4 \oplus x_5, \quad x'_{\omega 2} = x_3 \oplus x_4 \oplus x_5 \oplus x_6, \quad x'_{\omega 3} = x_3, \quad x'_{\omega 4} = x_2 \oplus x_3 \oplus x_5, \quad x'_{\omega 5} = x_4,$$

$$x'_{\omega 6} = x_1 \oplus x_2, \quad x'_{\omega 7} = x_4 \oplus x_6, \quad x'_{\omega 8} = x_2 \oplus x_3 \oplus x_4 \oplus x_6, \quad x'_{\omega 9} = x_3 \oplus x_5 \oplus x_6,$$

When determining the next argument ω^j , $j = 1 \div 9$ in the process of synthesis, the values of the signal $x'_{\omega j}$ inversion are also determined for each tangent layer - $\eta_j^{(00)}, \eta_j^{(01)}, \eta_j^{(10)}, \eta_j^{(11)}$. The combined version of all inversions is represented by the matrix structure below

$$\begin{array}{l} j = \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \\ \eta_j^{(00)} = \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \\ \eta_j^{(01)} = \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \\ \eta_j^{(10)} = \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \\ \eta_j^{(11)} = \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \end{array}$$

Single values show the required connections in matrix **OR** for the first input j of the adder $m2$, $j = 1 \div 9$, with the corresponding outputs of the decoder **DC**.

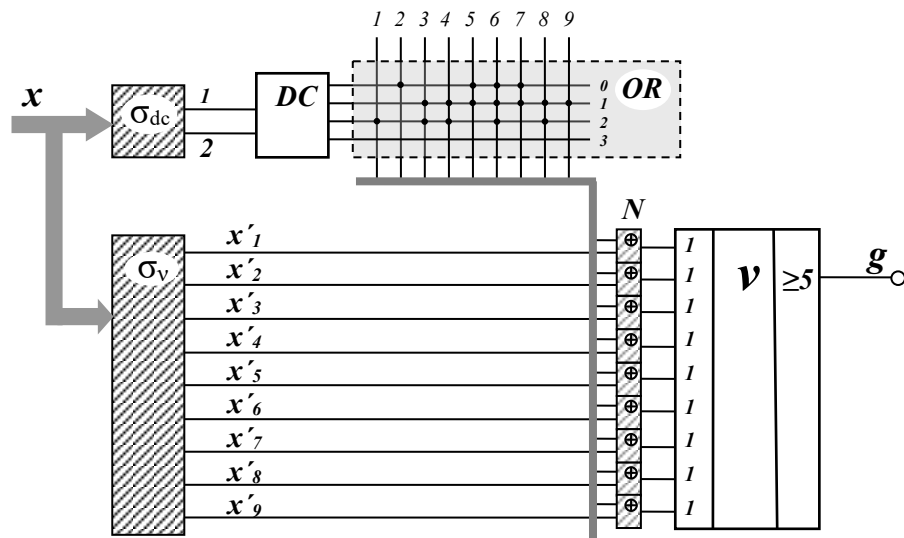


Fig. 3. A scheme of a hyperneuron module implementing the approximating function $g(x)$.

Hyperneuron settings on the performing approximating function $g(x)$ are determined by means of choosing linear transformations σ_v , σ_{dc} and vectors of the inversion $\eta^{(00)} \div \eta^{(11)}$. Therefore, all the weights a_j of the neuron v inputs are assigned equal to $a_j = 1$. Synthesized for this example, structure of hyperneuron is depicted in Figure 3.

Table 1 gives distribution of single neurons k_1 and zero k_0 values of the function f in the tangent layers corresponding to levels w_g hyperneuron excitation. Table 2 shows the corresponding overall distribution. For the convenience of perception, the values corresponding to levels w_g of the neuron's excitation, equal or exceeding the threshold value $w_g(x) \geq 5$ are highlighted in grey.

Table 1

	w_g	0	1	2	3	4	5	6	7	8	9
layer 00	k_1	–	–	1	–	–	3	2	2	–	–
	k_0	–	–	1	2	4	1	–	–	–	–
layer 01	k_1	–	–	–	–	–	4	2	2	–	–
	k_0	–	–	2	2	4	–	–	–	–	–
layer 10	k_1	–	–	–	–	–	4	2	2	–	–
	k_0	–	–	2	2	4	–	–	–	–	–
layer 11	k_1	–	–	–	–	–	2	3	1	1	–
	k_0	–	1	1	3	3	1	–	–	–	–

Table 2

	w_g	0	1	2	3	4	5	6	7	8	9
Σ	k_1	–	–	1	–	–	13	9	7	1	–
	k_0	–	1	6	9	15	2	–	–	–	–

Karnaugh tables are shown in Fig.4 in order to illustrate the threshold logic of hyperneuron function formation.

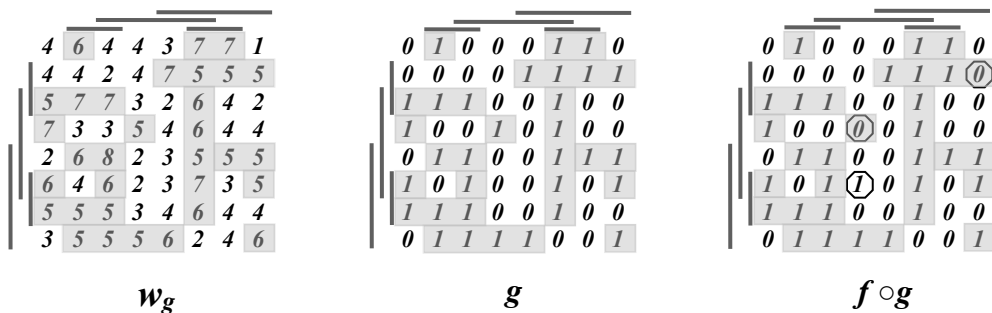


Fig. 4. Karnaugh tables displaying the hyperneuron synthesis results:
 w_g – matrix $w_g(x)$ of hyperneuron excitation level according to the input signal x ;
 g – a function realized by the hyperneuron;
 $f \circ g$ – function f with highlighted values different from g values.

Table w_g displays the common for tangent layers of neuron excitation levels distribution $w_g(x)$, Table g shows the approximating function $g(x)$ of the neuron, and in Table $f \circ g$ the comparison of the initial and the approximating function is presented. In all three tables, the grey background highlights the values of argument x , for which $g(x) = 1$. The table $f \circ g$ demonstrates the presence of differences in functions f and g for three values of the argument x . The numerical estimation of the approximation result, where approximation index is q , is defined here by the expression $q = |f| - |f \oplus g|$, where $|f|$ and $|f \oplus g|$ denote the ranks of the corresponding functions. In the example under consideration, index value is $q = 31 - 3 = 28$.

It is clear that for the given function f additional technical costs to the neuron, determined by the linear block σ_v and **OR** matrix seem to be excessive. However, as it has been noted in [3], when the dimension n increases, the complexity of LF and the implementing them circuits rises more than exponentially, while the complexity of the linear block, that simplifies the implementing schemes greatly, increases linearly. This fact is also reflected in the efficiency of the hyperneuron structure for the functions with dimension $n \geq 10$. So, for one the randomly assigned LF from 16 variables, having the number of single values $k = 32766$, the structure of the approximating hyperneuron has the following blocks: decoder **DC** with the dimension 2×4 , a linear block $\sigma_v - 19 \times 16$, matrix **OR** – 4×19 . In this case, the value of the approximation index reaches $q = 2322$.

5. CONCLUSIONS

The propose neural construction contains blocks with matrix structure, and the synthesis of the construction, its adjustment is accompanied by the calculations, using matrix operations. At each stage of the calculations Walsh spectral representations are used in order to ensure a quantitative assessment of the efficiency that have been made. As a result, the expenditure on variants selection is significantly reduced, and the way of optimizing the construction is becoming clear.

An increase in the approximation accuracy as regards the given function is carried out in the hyperneuron module design by changing the number t inputs of the linearized decoder and the number m inputs of the neuron v . The optimal choice of options is determined by the comparison of the resulting approximation index q with the corresponding volume of matrix blocks in the construction.

If we have a case with large dimensions, then a cascade scheme is quite applicable. For example, it is quite acceptable to perform a general preliminary stratification of a linear space, followed by the synthesis of a hyperneuron for each individual layer.

The design as a whole will be a logical mechanism with a well-analyzed structure.

REFERENCES

1. Sychev, AN 1981, 'Adaptive Algorithm for Synthesizing Linear Threshold Networks', *Avtomatika i vychislitel'naja tekhnika*, no. 5, pp. 17-24.
2. Sychev, AN 2014, 'Optimization of logical switching networks in neurostructures', *Automatic Control and Computer Sciences*, vol. 48, no. 4, pp. 196-206.
3. Karpovskii, MG & Moskalev, ES 1973, *Spectral Methods of Analysis and Synthesis of Discrete Devices*, Energy, Leningrad.
4. Sychev, AN 2011, 'Logic Neuron Model Using Walsh Spectral Representations', *Automatic Control and Computer Sciences*, vol. 45, no. 5, pp. 241-258.